

WHAT IS CLAIMED IS:

- 1 1. A computer implemented method for selecting a code image during a
2 reboot routine, comprising:
3 maintaining multiple code images in a memory device;
4 executing a first operation routine;
5 incrementing a first counter if the first operation routine succeeds;
6 executing a second operation routine;
7 incrementing a second counter if the second operation routine succeeds; and
8 using the first and second counters to select one of the code images from the
9 memory device to execute.
- 1 2. The method of claim 1, further comprising:
2 designating one code image as non-operational if the first counter is a first value
3 and the second counter is a second value, wherein one other code image not designated as
4 non-operational is selected to execute.
- 1 3. The method of claim 2, wherein the first value is greater than zero and the
2 second value is zero.
- 1 4. The method of claim 2, further comprising:
2 receiving an update to the code image;
3 determining whether one code image is designated as non-operational; and
4 overwriting the code image designated as non-operational with the received
5 update to the code image if one code image is designated as non-operational.
- 1 5. The method of claim 4, further comprising:
2 determining an earliest version of the code images in the memory device; and

Sub A5

0375544-010504

3 overwriting the determined earliest version of the code image if one code image is
4 not designated as non-operational.

1 6. The method of claim 4, further comprising:
2 determining whether one code image is corrupted; and
3 if one code image is corrupted, overwriting the corrupted code image with the
4 received update before determining whether one code is non-operational.

1 7. The method of claim 1, wherein the first operation routine comprises a
2 reboot routine and the second operation routine comprises an initialization routine.

1 8. The method of claim 7, further comprising:
2 incrementing the second counter if the initialization routine successfully
3 completed;
4 rebooting if the initialization routine failed; and
5 performing another iteration of all previous steps after rebooting.

1 9. The method of claim 7, further comprising:
2 selecting one copy of the code image, wherein the executed initialization routine
3 is a component of the selected code image, wherein the selected code image is designated
4 as non-operational if the first counter is the first value and the second counter is the
5 second value; and
6 selecting one other copy of the code image if the selected code image is
7 designated as non-operational.

1 10. The method of claim 1, wherein the code image comprise different
2 versions of the code image.

Sub A5

RECEIVED

1 11. The method of claim 1, wherein the first operation routine comprises a
2 reboot routine and the second operation routine comprises an initialization routine, and
3 wherein the code images include a function routine to perform an operation after
4 initialization, further comprising:

5 executing the function routine in one code image;
6 incrementing a third counter associated with the code image including the
7 executed function routine if the function routine succeeded; and
8 using the third counter, in addition to the first and second counters, to select one
9 of the multiple copies of the code image from the memory device to execute.

1 12. The method of claim 11, further comprising:
2 designating one code image as operational if the first, second, and third counters
3 satisfy at least one threshold value, wherein the code image designated as operational is
4 automatically selected from the memory device to execute after subsequent reboot
5 operations.

1 13. The method of claim 11, further comprising:
2 designating one code image as non-operational if the first, second, and third
3 counters satisfy at least one threshold value, wherein one other code image not designated
4 as non-operational is selected from the memory device and executed.

1 14. The method of claim 13, wherein the threshold value for the third counter
2 is zero and wherein the at least one threshold value for the first and second counters is
3 greater than zero

1 15. The method of claim 13, further comprising:
2 incrementing the second counter if the initialization routine successfully
3 completed;

Sub A5

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

4 rebooting if the initialization or function routine failed; and
5 performing another iteration of all previous steps after rebooting.

1 16. The method of claim 11, wherein the code image includes multiple
2 function routines, wherein there is one counter for each of the multiple function routines,
3 and further comprising:
4 designating one code image as operational if the first counter, second counter, and
5 each counter associated with a function routine satisfy at least one threshold value,
6 wherein the code image designated as operational is automatically selected from the
7 memory device to execute after subsequent reboot operations.

1 17. The method of claim 1, wherein one operation routine comprises one of a
2 reboot routine, an initialization routine or a function routine to perform a device specific
3 operation.

1 18. A computer system for selecting a code image during a reboot routine,
2 comprising:
3 a processor;
4 a memory device maintaining multiple code images, wherein the processor is
5 capable of accessing them memory device;
6 program logic executed by the processor, wherein the program logic causes the
7 processor to perform:
8 (i) executing a first operation routine;
9 (ii) incrementing a first counter if the first operation routine succeeds;
10 executing a second operation routine;
11 (iii) incrementing a second counter if the second operation routine
12 succeeds; and

Sub A5

RECEIVED

13 (iv) using the first and second counters to select one of the code images
14 from the memory device to execute.

1 19. The system of claim 18, wherein the program logic is further capable of
2 causing the processor to perform:
3 designating one code image as non-operational if the first counter is a first value
4 and the second counter is a second value, wherein one other code image not designated as
5 non-operational is selected to execute.

1 20. The system of claim 19, wherein the first value is greater than zero and the
2 second value is zero.

1 21. The system of claim 19, wherein the program logic is further capable of
2 causing the processor to perform:
3 receiving an update to the code image;
4 determining whether one code image is designated as non-operational; and
5 overwriting the code image designated as non-operational with the received
6 update to the code image if one code image is designated as non-operational.

1 22. The system of claim 21, wherein the program logic is further capable of
2 causing the processor to perform:
3 determining an earliest version of the code images in the memory device; and
4 overwriting the determined earliest version of the code image if one code image is
5 not designated as non-operational.

1 23. The system of claim 21, wherein the program logic is further capable of
2 causing the processor to perform:
3 determining whether one code image is corrupted; and

Sub A5

FILED IN TUC920000052US1

24. The system of claim 18, wherein the first operation routine comprises a reboot routine and the second operation routine comprises an initialization routine.

```

3      incrementing the second counter if the initialization routine successfully
4      completed;

```

1 26. The system of claim 24, wherein the program logic is further capable of
2 causing the processor to perform:

7 selecting one other copy of the code image if the selected code image is
8 designated as non-operational.

28. The system of claim 18, wherein the first operation routine comprises a
reboot routine and the second operation routine comprises an initialization routine, and
wherein the code images include a function routine to perform an operation after

3 incrementing the second counter if the initialization routine successfully
4 completed;
5 rebooting if the initialization or function routine failed; and
6 performing another iteration of all previous steps after rebooting.

1 33. The system of claim 28, wherein the code image includes multiple
2 function routines, wherein there is one counter for each of the multiple function routines,
3 and further comprising:
4 designating one code image as operational if the first counter, second counter, and
5 each counter associated with a function routine satisfy at least one threshold value,
6 wherein the code image designated as operational is automatically selected from the
7 memory device to execute after subsequent reboot operations.

1 34. The system of claim 18, wherein one operation routine comprises one of a
2 reboot routine, an initialization routine or a function routine to perform a device specific
3 operation.

1 35. An article of manufacture for selecting a code image during a reboot
2 routine, wherein the article of manufacture includes code in a computer readable medium
3 capable of causing a processor to perform:
4 maintaining multiple code images;
5 executing a first operation routine;
6 incrementing a first counter if the first operation routine succeeds;
7 executing a second operation routine;
8 incrementing a second counter if the second operation routine succeeds; and
9 using the first and second counters to select one of the code images to execute.

Sub A5

FILED IN TUC920000052US1

1 36. The article of manufacture of claim 35, wherein the article of manufacture
2 code is further capable of causing the processor to perform:

3 designating one code image as non-operational if the first counter is a first value
4 and the second counter is a second value, wherein one other code image not designated as
5 non-operational is selected to execute.

1 37. The article of manufacture of claim 36, wherein the first value is greater
2 than zero and the second value is zero.

1 38. The article of manufacture of claim 36, wherein the article of manufacture
2 code is further capable of causing the processor to perform:

3 receiving an update to the code image;
4 determining whether one code image is designated as non-operational; and
5 overwriting the code image designated as non-operational with the received
6 update to the code image if one code image is designated as non-operational.

1 39. The article of manufacture of claim 38, wherein the article of manufacture
2 code is further capable of causing the processor to perform:

3 determining an earliest version of the code images; and
4 overwriting the determined earliest version of the code image if one code image is
5 not designated as non-operational.

1 40. The article of manufacture of claim 38, wherein the article of manufacture
2 code is further capable of causing the processor to perform:

3 determining whether one code image is corrupted; and
4 if one code image is corrupted, overwriting the corrupted code image with the
5 received update before determining whether one code is non-operational.

Sub A5

FILED IN 4385-1

1 41. The article of manufacture of claim 35, wherein the first operation routine
2 comprises a reboot routine and the second operation routine comprises an initialization
3 routine.

1 42. The article of manufacture of claim 41, wherein the article of manufacture
2 code is further capable of causing the processor to perform:
3 incrementing the second counter if the initialization routine successfully
4 completed;
5 rebooting if the initialization routine failed; and
6 performing another iteration of all previous steps after rebooting.

1 43. The article of manufacture of claim 41, wherein the article of manufacture
2 code is further capable of causing the processor to perform:
3 selecting one copy of the code image, wherein the executed initialization routine
4 is a component of the selected code image, wherein the selected code image is designated
5 as non-operational if the first counter is the first value and the second counter is the
6 second value; and
7 selecting one other copy of the code image if the selected code image is
8 designated as non-operational.

1 44. The article of manufacture of claim 35, wherein the code image comprise
2 different versions of the code image.

1 45. The article of manufacture of claim 35, wherein the first operation routine
2 comprises a reboot routine and the second operation routine comprises an initialization
3 routine, and wherein the code images include a function routine to perform an operation
4 after initialization, wherein the article of manufacture code is further capable of causing
5 the processor to perform

Sub A5

RECEIVED

Sub A5

Year	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100
1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	

SECRET

SECRET

[illegible]

SECRET

SECRET

SECRET

SECRET

1 50. The article of manufacture of claim 45, wherein the code image includes
2 multiple function routines, wherein there is one counter for each of the multiple function
3 routines, wherein the article of manufacture code is further capable of causing the
4 processor to perform:

5 designating one code image as operational if the first counter, second counter, and
6 each counter associated with a function routine satisfy at least one threshold value,
7 wherein the code image designated as operational is automatically selected to execute
8 after subsequent reboot operations.

1 51. The article of manufacture of claim 35, wherein one operation routine
2 comprises one of a reboot routine, an initialization routine or a function routine to
3 perform a device specific operation.

Add A57

Sub A5

RECEIVED